

### Abstract

Navier-Stokes 方程是一种简化的流体模型, 模拟自然界的流体, 从大尺度的海洋洋流, 到小尺度的血管流。深入研究这个流体模型, 有助于我们更好的了解自然。Navier-Stokes 方程中有非线性项, 并且当流速很大的时候, 求解会很困难, 并且在三维的情况下, 还会有湍流出现。因此, 研究 Navier-Stokes 方程的数值解法是有意义的。

本文主要是在嵌套的两层网格, 结合移动网格方法, 用低阶混合元方法来求解 Navier-Stokes 方程,

# 第 1 章 引言

众所周知, 混合元是求解不可压 Navier-Stokes 方程的一种重要方法。为了保证解的唯一性, 需要使速度和压力两个解的空间满足 LBB 条件。其中一种办法是使得速度空间相对压力空间来说, 自由度足够多, 例如 Taylor-Hood 元。另外一种方法是对压力空间施加一些约束, 比如说稳定化的 P1P1 元, P1P0 元。为了减少计算量, 提高计算效率, 通常会运用自适应网格方法。在文献 [1],[2] [3] 中, 运用了  $h$ -自适应的 P2P1 元。在实际的工程计算中, 我们通常倾向于使用线性元, 而非高次元。[4] 提出自适应网格和稳定化 P1P1 元、P1P0 元相结合。[5] 将移动网格方法应用到不可压 Navier-Stokes 方程的求解。我们移动网格部分的策略是基于文献 [5] 中的工作。然而, 不稳定的有限元对应用到  $h$ -自适应上有一定的技术困难。综合以上考虑, 我们选取稳定的 P1ISOP2P1 元, 它自然满足 LBB 条件, 详细见 [6]。在 P1ISOP2P1 元, 速度单元所在的网格可以有压力单元的网格加密一次得到, 如 Figure (1.1) 注意到在速度单元  $u$  和压力单元  $p$  上均是线性元。但是, 速度单元网格上的 6 个基函数不在同一个速度单元上。因此在拼装散度块矩阵, 即拼装压力单元和速度单元, 这个过程并不显然。

在本文的工作中, 我们选取 4P1-P1 元, 它与 ISOP2P1 拥有相同的网格结构, 自然也满足 inf-sup 条件, 但需要指出的是速度单元上的基函数都局部的位于相同的速度单元上。这给我们拼装散度块矩阵提供了便利: 只要我们建立四个速度单元和大的压力单元之间的索引。索引的建立又依赖于两层网格的数据结构, 这种网格结构与 [7] 中提出的几何遗传树是一致的。

基于这种网格遗传树的结构, 我们可以很容易的建立四个小的速度单元与压力宏单元间的索引。同时在 [7] 中  $h$ -自适应方法也是基于这种树结构。自然的, 我们可以将  $h$ -自适应方法应用到 4P1-P1 元上。自适应网格可以减少计算量, 同时可以研究流体局部小尺度上的现象, 比如涡。同时, 移动网格方法也可以应用在 4P1-P1 元上, 网格移动我们只移动压力网格, 当压力网格移动完, 速度网格由压力网格加密一次即可得到。但是需要注意的是, 求解不可压的 Navier-Stokes 方程, 在将旧网格的解插值到移动后的网格的过程, 要满足散度为 0 的条件, 这个工作由 [5] 中给出。

本文的框架组织如下: 在第二章中, 我们介绍 Navier-Stokes 方程的知识。在第三章中我们展示 4P1-P1 的数据结构, 接下来我们用 4P1-P1 元近似 Navier-Stokes 方程。移动网格的策略将在第五章中给出。最后, 我们给出数值例子。

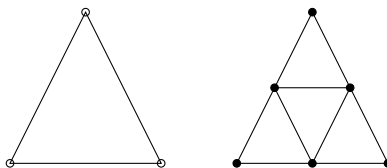


Figure 1.1: 左: 压力  $p$  单元,  $\circ$  为  $p$  单元上的自由度; 右: 4 个速度  $v$  单元,  $\bullet$  表示  $v$  单元上的自由度.

## 第 2 章 流体方程

无量纲化的不可压 Navier-Stokes 方程如下：

$$\begin{aligned}\frac{\partial \vec{u}}{\partial t} - \nu \nabla^2 \vec{u} + \vec{u} \cdot \nabla \vec{u} + \nabla p &= \vec{f}, \\ \nabla \cdot \vec{u} &= 0,\end{aligned}\tag{2.1}$$

其中  $\nu$  为粘性系数，求解区域为  $\Omega$ 。系统 (2.1) 的初边值问题如下：

$$\begin{aligned}\vec{u} &= \vec{w}, & \text{on } \partial\Omega_D \times [0, T] \\ \nu \frac{\partial \vec{u}}{\partial n} - p &= \vec{0}, & \text{on } \partial\Omega_N \times [0, T], \\ \vec{u}|_{t=0} &= \vec{u}_0, & \text{in } \Omega.\end{aligned}\tag{2.2}$$

其中边界  $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ ， $\vec{n}$  表示边界的外法向。令  $L$  表示区域  $\Omega$  的特征长度， $U$  是入流边界上的最大流速，那么雷诺数定义为：

$$Re := \frac{UL}{\nu}.\tag{2.3}$$

### 第 3 章 数据结构

4P1-P1 是基于两套不同的网格和两种有限元空间。速度网格可以由压力网格全局加密一次得到。网格的数据结构是基于 [7] 中的几何遗传树结构, 如 Figure(3.1) 所示。一个宏压力单元对应这四个速度单元, 通过遍历一次所有的速度单元, 利用几何遗传树结构, 就可以建立速度单元和压力单元的 1-1 对应, 如 algorithmic(1)。索引建立后, 我们用 P1 元来拼装散度矩阵。

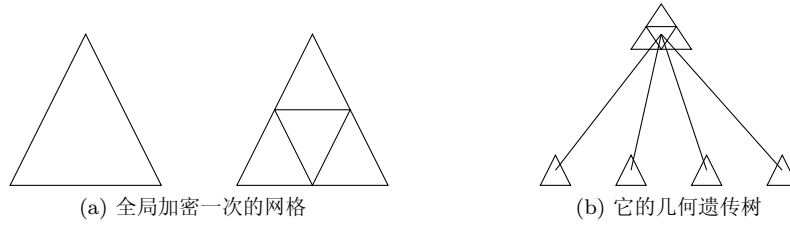


Figure 3.1: 几何遗传树结构

---

**Algorithm 1** 构建速度单元和压力单元间的索引

---

```

1: achieveiterator  $\leftarrow$  irregualerMeshV.beginActiveElement()
2: enditerator  $\leftarrow$  irregualerMeshV.endActiveElement()
3: while achieveiterator  $\neq$  enditerator do
4:   int index-v-element  $\leftarrow$  achieveiterator-} index
5:   HElement  $\langle$  DIM, DIM  $\rangle$ * parent  $\leftarrow$  activeiterator-} parent
6:   int index-p-element  $\leftarrow$  parent-} index
7:   int n-child  $\leftarrow$  parent-} n-child
8:   index-p2v[index-p-element].resize(n-child)
9:   while  $i \geq 0$  and  $i < n-child$  do
10:    HElement  $\langle$  DIM, DIM  $\rangle$  *chi  $\leftarrow$  parent-}child[i]
11:    int index-v-element  $\leftarrow$  child-}index
12:    index-p2v[index-p-element][i]  $\leftarrow$  index-v-element
13:    index-v2p[index-v-element]  $\leftarrow$  index-p-element
14:   end while
15: end while

```

---

## 第 4 章 有限元近似

我们定义解和检验空间如下：

$$\mathbf{H}_E^1 := \{ \vec{u} \in \mathcal{H}^1(\Omega)^d \mid \vec{u} = \vec{w} \text{ on } \partial\Omega_D \}, \quad (4.1)$$

$$\mathbf{H}_{E_0}^1 := \{ \vec{v} \in \mathcal{H}^1(\Omega)^d \mid \vec{v} = \vec{0} \text{ on } \partial\Omega_D \}, \quad (4.2)$$

那么变分形式为：寻找  $(\vec{u}, p) \in (\mathbf{H}_E^1, L_2(\Omega))$  使得

$$\int_{\Omega} \frac{\partial \vec{u}}{\partial t} \cdot \vec{v} + \nu \int_{\Omega} \nabla \vec{u} : \nabla \vec{v} + \int_{\Omega} (\vec{u} \cdot \nabla \vec{u}) \cdot \vec{v} - \int_{\Omega} p (\nabla \cdot \vec{v}) = \int_{\Omega} \vec{f} \cdot \vec{v}, \quad (4.3)$$

$$\forall \vec{v} \in \mathbf{H}_{E_0}^1,$$

$$\int_{\Omega} q (\nabla \cdot \vec{u}) = 0, \quad (4.4)$$

$$\forall q \in L_2(\Omega).$$

其中  $\nabla \vec{u} : \nabla \vec{v}$  表示纯量的乘积，在二维中为  $\nabla u_x \cdot \nabla v_x + \nabla u_y \cdot \nabla v_y$ .

假设  $\tau_h$  是  $\Omega$  上对压力网格的三角剖分，网格尺度  $h = \max_{T \in \tau_h} \text{diam}(T)$ ,  $T$  为三角剖分  $\tau_h$  的单元。对应的， $\tau_{\frac{h}{2}}$  是对速度网格的三角剖分。基于  $\tau_{\frac{h}{2}}$  和  $\tau_h$  上的有限元空间  $X_E^h$  和  $P_h$  满足

$$X_E^h \subset \mathcal{H}_E, \quad P_h \subset L_2(\Omega)$$

那么 (4.3) 和 (4.4) 可以写成如下形式：寻找  $(\vec{u}_h, p_h) \in X_E^h \times P_h$  使得

$$\begin{aligned} & \int_{\Omega} \frac{\partial \vec{u}_h}{\partial t} \cdot \vec{v}_h + \nu \int_{\Omega} \nabla \vec{u}_h : \nabla \vec{v}_h \\ & + \int_{\Omega} (\vec{u}_h \cdot \nabla \vec{u}_h) \cdot \vec{v}_h - \int_{\Omega} p_h (\nabla \cdot \vec{v}_h) = \int_{\Omega} \vec{f} \cdot \vec{v}_h, \quad \forall \vec{v}_h \in \mathbf{X}_0^h; \quad (4.5) \\ & \int_{\Omega} q_h (\nabla \cdot \vec{u}_h) = 0, \quad \forall q_h \in P^h. \end{aligned}$$

时间方向至少三阶 Runge-Kutta 方法才能保证数值稳定性。为了简便，我们这里只用显示 Euler 格式：  $\forall (\vec{v}_h, q_h) \in \mathbf{X}_0^h \times P^h$

$$\begin{aligned} & \int_{\Omega} \frac{\vec{u}_h^{(n+1)} - \vec{u}_h^{(n)}}{\delta t} + \nu \int_{\Omega} \nabla \vec{u}_h^{(n+1)} : \nabla \vec{v}_h - \int_{\Omega} p_h^{(n+1)} (\nabla \cdot \vec{v}_h) = \int_{\Omega} (\vec{u}_h^{(n)} \cdot \nabla \vec{u}_h^{(n)}) \\ & \int_{\Omega} q_h \nabla \cdot \vec{u}_h^{(n+1)} = 0. \end{aligned} \quad (4.6)$$

令  $\{\phi_j\}_{j=1}^n$  和  $\{\psi_k\}_{k=1}^m$  分别为速度和压力的线性元基函数。则数值解  $\vec{u}_h^{(n+1)} = (u_{xh}^{(n+1)}, u_{yh}^{(n+1)})$ ,  $p_h$  可以写成如下形式：

$$u_{xh}^{(n+1)} = \sum_{j=1}^n u_j \phi_j, \quad u_{yh}^{(n+1)} = \sum_{j=1}^n v_j \phi_j, \quad p_h^{(n+1)} = \sum_{k=1}^m p_k \psi_k$$

将  $u_{xh}^{(n+1)}, u_{yh}^{(n+1)}, p_h^{(n+1)}$  带入离散弱形式 (4.6) 中, 可以得到线性方程组

$$\begin{bmatrix} \frac{1}{dt}M + \nu A & 0 & B_x^T \\ 0 & \frac{1}{dt}M + \nu A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ g \end{bmatrix}, \quad (4.7)$$

其中  $M$  是  $n \times n$  的质量矩阵,  $A$  是拉普拉斯矩阵, 由以下形式:

$$\begin{aligned} A &= [a_{ij}], \quad a_{ij} = \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \\ M &= [m_{ij}], \quad m_{ij} = \int_{\Omega} \phi_i \phi_j \\ B_x^T &= [bx_{ik}^T], \quad bx_{ik}^T = \int_{\Omega} \psi_k \frac{\partial \phi_i}{\partial x} \\ B_y^T &= [by_{ik}^T], \quad by_{ik}^T = \int_{\Omega} \psi_k \frac{\partial \phi_i}{\partial y} \\ f_x &= [f_i], \quad f_i = \int_{\Omega} \left( \frac{u_{xh}^{(n)}}{dt} - (u_{xh}^{(n)} \frac{\partial u_{xh}^{(n)}}{\partial x} + u_{yh}^{(n)} \frac{\partial u_{xh}^{(n)}}{\partial y}) \right) \phi_i \\ f_y &= [f_i], \quad f_i = \int_{\Omega} \left( \frac{u_{yh}^{(n)}}{dt} - (u_{xh}^{(n)} \frac{\partial u_{yh}^{(n)}}{\partial x} + u_{yh}^{(n)} \frac{\partial u_{yh}^{(n)}}{\partial y}) \right) \phi_i \\ g &= [g_k], \quad g_i = 0. \end{aligned}$$

接下来我们着重解释一下散度块矩阵  $B_x B_y, B_x^T, B_y^T$  的拼装。以  $B_x^T$  为例, 令  $\Delta_{v_i}$  为一个速度单元, 我们可以通过上一节建立的单元索引来找到对应的压力单元  $\Delta_{p_k}$  (如 Figure (4.1)).  $\phi_{i_1}, \phi_{i_2}, \phi_{i_3}$  是定义在速度单元  $\Delta_{v_i}$  上的线性元基函数, 下标  $i_1, i_2, i_3$  表示该顶点上的自由度在速度全部自由度的全局编号。同时  $\psi_{k_1}, \psi_{k_2}, \psi_{k_3}$  是定义在压力单元  $\Delta_{p_k}$  上的线性元基函数,  $k_1, k_2, k_3$  代表单元  $\Delta_{p_k}$  上的自由度在压力全部自由度中的全局编号。因此单元  $\Delta_{p_k}$  和  $\Delta_{v_i}$  对  $B_x^T$  的贡献为

$$\begin{bmatrix} \int_{\Delta_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_1}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_1}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_1}}{\partial x} \\ \int_{\Delta_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_2}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_2}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_2}}{\partial x} \\ \int_{\Delta_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_3}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_3}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_3}}{\partial x} \end{bmatrix} \quad (4.8)$$

将贡献矩阵 (4.8) 加到  $B_x^T$  中的相应位置  $(i_1, k_1), (i_1, k_2), (i_1, k_3), (i_2, k_1), (i_2, k_2), (i_2, k_3), (i_3, k_1), (i_3, k_2), (i_3, k_3)$  to  $B_x^T$  上。通过遍历所有的速度单元, 重复上面的操作,  $B_x^T$  拼装完成。  $B_y^T$  也可以同样拼装。

$B_x$  和  $B_y$  的拼装是遍历所有的压力单元, 通过单元索引, 找到对应的四个速度单元, 然后分别用压力单元和四个速度单元进行拼装。过程相似, 便不再赘述。

我们主要的想法是有两套网格, 速度网格和压力网格, 速度网格可以由压力网格加密一次得到。因此我们对压力网格进行移动, 然后对移动完的压力网格全局加密一次, 即可实现对速度网格的移动。为了更好的说明我们的数值方法, 我们将算法的流程图如 Algorithm(2) 所示。

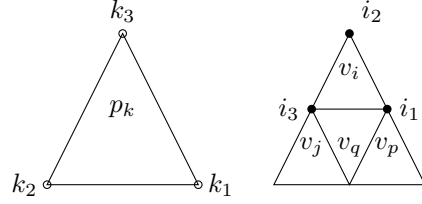


Figure 4.1: 左:  $p$  单元; 右: 与压力单元对应的 4 个速度  $v$  单元

---

**Algorithm 2** 移动网格方法来求解 Navier Stokes 方程

---

- 1: **while**  $t_n < T$  **do**
  - 2:   在速度网格  $\Delta_v^{(n)}$  和压力网格  $\Delta_p^{(n)}$  上, 求解  $t = t_n$  时刻的 Navier-Stokes 方程 (4.7), 获得数值解  $\vec{u}_h^{(n)}, p_h^{(n)}$ .
  - 3:   在压力网格  $\Delta_p^{(n)}$  上, 用  $\vec{u}_h^{(n)}, p_h^{(n)}$  来计算控制函数, 并通过求解 (5.6), 获得  $\vec{\xi}^*$ .
  - 4:   判断  $\|\vec{\xi}^* - \vec{\xi}^{(0)}\|_{L^2}$  是否小于容忍量  $\epsilon$ , 如果是迭代结束, 否则, 继续做 5 - 8.
  - 5:   用  $\vec{\xi}^* - \vec{\xi}^{(0)}$  两者之差来计算网格  $\Delta_p^{(n)}$  的移动量  $\delta\vec{x}$ .
  - 6:   利用 4 中  $\delta\vec{x}$ , 在速度网格  $\Delta_v^{(n)}$  上求解更新数值解的方程 (5.19), 得到新网格上的中间量  $\vec{u}_{h,*}^{(n)}, p_{h,*}^{(n)}$ .
  - 7:   更新  $\Delta_p^{(n)}$ , 通过几何遗传树结构, 来同步  $\Delta_v^{(n)}$ , 得到新的  $\Delta_p^{(n+1)}$  和  $\Delta_v^{(n+1)}$ .
  - 8:   回到 3.
  - 9:   在  $\Delta_v^{(n+1)}$  和  $\Delta_p^{(n+1)}$  上求解 Navier-Stokes 方程 (4.7). 从而真正获得  $t = t_{n+1}$  时刻的数值解  $\vec{u}_h^{(n+1)}, p_h^{(n+1)}$ .
  - 10:    $n = n + 1$
  - 11: **end while**
-

## 第 5 章 移动网格策略

在  $t = t_{n+1}$  时刻, 用上一章的方法可以得到有限元解  $(\vec{u}_h^{(n+1)}, p_h^{(n+1)})$ . 下面的问题是如何用新的数值解和旧的网格  $\mathcal{T}_h^{(n)}$  来获得新的网格  $\mathcal{T}_h^{(n+1)}$ . 我们采取文献 [5] 中的方法, 注意到我们区域的边界均是 Dirichlet 边界, 分为以下四步:

### 5.1 Step 1 获取 Monitor

选择一个合适的控制函数, 对于移动网格的结果是非常重要的。应用在不可压 Navier-Stokes 方程上的控制函数主要有如下几种. 令  $m = \frac{1}{G}$  其中  $m$  是 (5.6) 中的一个纯量函数。关于  $G$  有集中不同的选择. 一种是基于涡量

$$G_0 = \sqrt{1 + \alpha|\omega|^\beta} \quad (5.1)$$

其中  $\omega = \nabla \times \vec{u}$ ,  $\alpha$  和  $\beta$  是两个正的常数。

另外一个选择,  $G$  是基于数值解的梯度

$$G_1 = \sqrt{1 + \alpha|\nabla \vec{u}|^\beta} \quad (5.2)$$

对于线性元  $v_h$  逼近真实解  $v$ , 下面的后验误差估计公式可以用来近似计算误差

$$|v - v_h|_{1,\Omega} \sim \eta(v_h) := \sqrt{\sum_{l: \text{内部边界}} \int_l [\nabla v_h \cdot \vec{n}_l]^2 dl} \quad (5.3)$$

其中  $[\cdot]_l$  意味着边  $l$  上的跳跃, 即  $[v]_l = v|_{l^+} - v|_{l^-}$ . 很自然的在每个单元上等分布数值误差  $\eta(v_h)$ , 控制函数为一下形式

$$G_2 = \sqrt{1 + \alpha\eta^2(v_h)} \quad (5.4)$$

[5] 中对 (5.4) 进行了改进

$$G_3 = \sqrt{1 + \alpha [\eta(v_h)/\max \eta(v_h)]^\beta} \quad (5.5)$$

其中  $\beta > 2$  时有更好的效果。

### 5.2 Step 2 获取新的逻辑网格

求解椭圆形方程

$$\begin{aligned} \nabla_{\vec{x}} (m \nabla_{\vec{x}} \vec{\xi}) &= 0 \\ \vec{\xi} &= \vec{\xi}_b \end{aligned} \quad (5.6)$$



其中  $m$  是上一节中的纯量函数, 通常依赖于  $(\bar{u}_h^{(n+1)}, p_h^{(n+1)})$ 。我们定义初始的逻辑网格  $\mathcal{T}_c(\mathcal{A}^0$  为它的节点)。一旦初始的逻辑网格给定, 在整个的求解过程中, 将一直保持不变。通过求解 (5.6) 我们可以得到新的逻辑网格  $\mathcal{T}_c^*(\mathcal{A}^*$  为它的节点)。

### 5.3 Step 3 物理网络的移动方向

我们先引入一些定义。  $\mathcal{T}_h$  为物理区域上的三角剖分。第  $i$  个点定义为  $X_i$ , 以  $X_i$  为顶点的单元的集合称之为  $T_i$ 。相应的计算区域上的标记为  $\mathcal{T}_c, \mathcal{A}_i$  和  $T_{i,c}$ 。  $\mathcal{A}_i$  点在计算区域上的坐标定义为  $(\mathcal{A}_i^1, \mathcal{A}_i^2)^T$ 。在 Step 1 结束后, 我们得到了新的逻辑网格  $\mathcal{T}_c^*$  和它的顶点  $\mathcal{A}_i^*$ 。从而我们得到新旧逻辑网络的差:

$$\delta \mathcal{A}_i = \mathcal{A}^{(0)} - \mathcal{A}_i^* \quad (5.7)$$

对于一个给定的单元  $E \in \mathcal{T}_h$ ,  $X_{E_k}, 0 \leq k \leq 2$ , 作为它的三个顶点。从  $V_{\mathcal{T}_c^*}(\Omega)$  到  $V_{\mathcal{T}}(\Omega)$  的分片线性映射在单元  $E$  上的梯度是常数, 并且满足下面的方程组:

$$\begin{pmatrix} \mathcal{A}_{E_1}^{*,1} - \mathcal{A}_{E_0}^{*,1} & \mathcal{A}_{E_2}^{*,1} - \mathcal{A}_{E_0}^{*,1} \\ \mathcal{A}_{E_1}^{*,2} - \mathcal{A}_{E_0}^{*,2} & \mathcal{A}_{E_2}^{*,2} - \mathcal{A}_{E_0}^{*,2} \end{pmatrix} \begin{pmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} \end{pmatrix} \\ = \begin{pmatrix} X_{E_1}^1 - X_{E_0}^1 & X_{E_2}^1 - X_{E_0}^1 \\ X_{E_1}^2 - X_{E_0}^2 & X_{E_2}^2 - X_{E_0}^2 \end{pmatrix}$$

求解上面的方程组, 可以获得单元  $E$  上的  $\partial \bar{x} / \partial \xi$ 。如果以单元的面积作为权重, 则第  $i$  个点的加权平均的位移定义如下:

$$\delta X_i = \frac{\sum_{E \in T_i} |E| \frac{\partial \bar{x}}{\partial \xi} |_{\text{in } E} \delta \mathcal{A}_i}{\sum_{E \in T_i} |E|}. \quad (5.8)$$

其中  $|E|$  代表单元  $E$  的面积。为了避免网格发生缠结, 在网格移动向量前乘上一个常量  $\mu$ , 即物理区域上新网格  $\mathcal{T}^*$  的节点表示为:

$$X_i^* = X_i + \mu \delta X_i. \quad (5.9)$$

文献 [5] 中提出  $\mu$  按以下方式给出:

$$\begin{vmatrix} 1 & 1 & 1 \\ x_0^1 + \mu \delta x_0^1 & x_1^1 + \mu \delta x_1^1 & x_2^1 + \mu \delta x_2^1 \\ x_0^2 + \mu \delta x_0^2 & x_1^2 + \mu \delta x_1^2 & x_2^2 + \mu \delta x_2^2 \end{vmatrix} = 0 \quad (5.10)$$

其中  $\bar{x}_i = (x_i^1, x_i^2), 0 \leq i \leq 2$  表示第  $i$  个点的坐标。令  $\mu_i^*$  为方程 (5.10) 的最小正根, 则令

$$\mu = \min(1, \frac{\mu_i^*}{2}). \quad (5.11)$$

## 5.4 Step 4 散度为 0 的插值

用移动网格方法求解不可压流体时, 要保证插值的过程散度是为 0 的。通过求解一个对流方程, 对流的速度是网格的移动速度, 从而实现旧的物理网格上的数值解到新的物理网格上数值解的插值。令  $u_h = \sum u_i \phi_i, u_h \in \mathcal{X}_E^h$ ,  $\phi_i$  是有限元空间  $\mathcal{X}_h$  的基函数。引入一个虚拟的时间  $\tau$ , 假设基函数  $\phi_i$  和  $u_i$  均是关于  $\tau$  的函数, 即  $\phi_i = \phi_i(\tau), u_i = u_i(\tau)$ 。我们引入一个从旧网格  $x^{\text{旧}}$  到新网格  $x^{\text{新}}$  网格点的连续变换:

$$x_i(\tau) = X_i + \tau(X_i^* - X_i), \quad \tau \in [0, 1] \quad (5.12)$$

其中  $X_i^* = x_i^{\text{新}}, X_i = x_i^{\text{旧}}$  基于 (5.12) 的连续形式  $x(\tau) = x_{\text{旧}} + \tau(x^{\text{新}} - x^{\text{旧}})$ , 基函数可以定义为  $\phi_i(\tau) = \phi_i(x(\tau))$  并且  $u_i = u_i(x(\tau))$ 。

在插值的过程中, 我们要保持解曲线  $u_h = \sum u_i \phi_i$  关于  $\tau$  在弱形式下是不变的。即对  $\forall \psi \in \mathcal{X}_h, (\partial_\tau u_h, \psi) = 0$ 。通过直接计算可得

$$\frac{\partial \phi}{\partial \tau} = -\nabla_{\vec{x}} \phi_i \cdot \delta \vec{x} \quad (5.13)$$

其中  $\delta \vec{x} = x^{\text{旧}} - x^{\text{新}}$ 。紧接着

$$\begin{aligned} 0 &= (\partial_\tau u_h, \psi) \\ &= (\partial_\tau \sum u_i(x(\tau)) \phi_i, \psi) \\ &= (\sum \phi_i \partial_\tau u_i(x(\tau)) + \sum u_i \partial_\tau \phi_i) \\ &= (\sum \phi_i \partial_\tau u_i(x(\tau)) - \sum u_i \nabla_{\vec{x}} \phi_i \cdot \delta \vec{x}, \psi) \\ &= (\sum \phi_i \partial_\tau u_i(x(\tau)) - \nabla_{\vec{x}} u_h \cdot \delta \vec{x}, \psi) \end{aligned} \quad (5.14)$$

我们将 (5.14) 应用到不可压流上, 即速度场要满足散度为 0 的条件。令  $\mathcal{X}_h$  为散度为 0 的空间:

$$\mathcal{X}_E^h = X_E^h \cap \{\vec{u}_h | \nabla \cdot \vec{u}_h = 0\} \quad (5.15)$$

那么 (5.14) 将变成: 寻找  $w_h \in \mathcal{X}_h$  使得

$$\left( \sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\vec{x}} \phi_i \cdot \delta \vec{x}, z_h \right) = 0 \quad \forall z_h \in \mathcal{X}_h. \quad (5.16)$$

上面的结果意味着

$$\sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\vec{x}} \phi_i \cdot \delta \vec{x} \in \mathcal{X}_h^\perp \quad (5.17)$$

其中  $\mathcal{X}_h^\perp + \mathcal{X}_h = L^2$ 。根据文献 [8] 中的定理 2.7, 如果区域  $\Omega$  是单连通的, 那么

$$\mathcal{X}_h^\perp = \{\nabla q | q \in H^1(\Omega)\} \quad (5.18)$$

则存在  $\nabla p \in \mathcal{X}_h^\perp$  使得

$$\begin{aligned} \sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\vec{x}} \phi_i \cdot \delta \vec{x} &= -\nabla p \\ \nabla_{\vec{x}} \cdot u_h &= 0. \end{aligned} \quad (5.19)$$

**Remark 5.4.1.** 这里的  $p$  跟外部 *Navier-Stokes* 方程的解  $p$  不一致，只是一个辅助量。

(5.19) 的弱形式：寻找  $(\vec{u}_h, p_h) \in X_E^h \times P_h$  使得

$$\begin{aligned} \left( \sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\vec{x}} \phi_i \cdot \delta \vec{x}, v_h \right) &= (p_h, \nabla v_h), & \forall v_h \in X_E^h. \\ (\nabla_{\vec{x}} \cdot u_h, q_h) &= 0, & \forall q_h \in P_h \end{aligned} \quad (5.20)$$

(5.19) 和 (5.20) 的初值为在  $t = t_n$  时刻的网格上， $t = t_{n+1}$  时刻外部 *Navier-Stokes* 方程的解。

时间方向的离散我们暂时先用线性 Euler 方法：

$$\begin{aligned} \left( \frac{\sum \phi_i u_{i,*}^{(n)} - \sum \phi_i u_i^{(n)}}{\Delta \tau}, v_h \right) - \left( \sum u_i^{(n)} \nabla \phi_i \cdot \delta \vec{x}, v_h \right) &= (p_h^{(n)}, \nabla v_h). \\ (\nabla \cdot u_{h,*}^{(n)}, q_h) &= 0 \end{aligned} \quad (5.21)$$

注意到在做数值解插值的过程中，物理网格还没有发生移动，这时候基函数  $\phi$  仍然是  $t = t_n$  时刻网格上的基函数。所以  $u_{h,*}^{(n)} = \sum u_{i,*}^{(n)} \phi_i^{(n)}$ ， $u_h^{(n)} = \sum u_i^{(n)} \phi_i^{(n)}$  简单整理一下：

$$\begin{aligned} \left( \frac{u_{h,*}^{(n)} - u_h^{(n)}}{\Delta \tau}, v_h \right) - (\nabla u_h^{(n)} \cdot \delta \vec{x}, v_h) &= (p_h^{(n)}, \nabla v_h). \\ (\nabla \cdot u_{h,*}^{(n)}, q_h) &= 0 \end{aligned} \quad (5.22)$$

其中  $u_h^{(n)}$  和  $p_h$  是在  $t = t_n$  时刻的网格上， $t = t_{n+1}$  时刻，*Navier-Stokes* 方程的数值解。而  $u_{h,*}^{(n)}$  和  $p_{h,*}^{(n)}$  是在新的网格上  $t = t_{n+1}$  时刻更新的解。但这组解不能当作外部 *Navier-Stokes* 方程的解。需要在新网格上重新求解 *Navier-Stokes* 方程，得到的解才是我们想要的解。

## 第 6 章 数值算例

### 6.1 Colliding Flow

这个例子为稳态 Stokes 方程的精确解，粘性系数  $\nu = 1.0$ :

$$u_x = 20xy^3; \quad u_y = 5x^4 - 5y^4; \quad p = 60x^2y - 20y^3 + \text{constant}. \quad (6.1)$$

其中计算区域  $\Omega = [-1, -1] \times [1, 1]$ , 边界条件全部是 Dirichlet 条件。这个例子是用来检验移动网格方法的收敛阶，此时解比较光滑。从文献 [6] 可知，我们期望移动网格的收敛阶：速度有二阶收敛，压力一阶收敛。我们先给出均匀网格时，误差的收敛阶，如 Table(6.1) 和 Table(6.2) 所示。

网格	$\ \vec{u} - \vec{u}_h\ _{L^2}$	误差阶	$\ \vec{u} - \vec{u}_h\ _{H^1}$	$\ p - p_h\ _{L^0}$	误差阶	$\ p - p_h\ _{H^1}$
$10 \times 10$	$1.42 \times 10^{-1}$		$3.65 \times 10^0$	$1.26 \times 10^0$		$2.06 \times 10^1$
$20 \times 20$	$3.54 \times 10^{-2}$	2.01	$1.81 \times 10^0$	$3.90 \times 10^{-1}$	1.62	$1.22 \times 10^1$
$40 \times 40$	$8.82 \times 10^{-3}$	2.01	$9.03 \times 10^{-1}$	$1.14 \times 10^{-1}$	1.71	$6.72 \times 10^0$
$80 \times 80$	$2.20 \times 10^{-3}$	2.00	$4.51 \times 10^{-1}$	$3.39 \times 10^{-2}$	1.68	$3.90 \times 10^0$

Table 6.1: 用均匀网格计算碰撞流的误差,  $\nu = 1.0$ .

网格	涡量 $L^2$ 误差	误差阶	散度 $L^2$ 误差	误差阶
$10 \times 10$	$2.62 \times 10^0$		$2.54 \times 10^0$	
$20 \times 20$	$1.31 \times 10^0$	1.00	$1.25 \times 10^0$	1.02
$40 \times 40$	$6.54 \times 10^{-1}$	1.00	$6.22 \times 10^{-1}$	1.00
$80 \times 80$	$3.27 \times 10^{-1}$	1.00	$3.10 \times 10^{-1}$	1.00

Table 6.2: 用均匀网格计算碰撞流的散度和涡量误差,  $\nu = 1.0$ .

网格	$\ \vec{u} - \vec{u}_h\ _{L^2}$	误差阶	$\ \vec{u} - \vec{u}_h\ _{H^1}$	$\ p - p_h\ _{L^0}$	误差阶	$\ p - p_h\ _{H^1}$
$10 \times 10$	$1.18 \times 10^{-1}$		$3.37 \times 10^0$	$8.49 \times 10^{-1}$		$1.84 \times 10^1$
$20 \times 20$	$2.94 \times 10^{-2}$	2.01	$1.67 \times 10^0$	$2.330 \times 10^{-1}$	1.82	$1.06 \times 10^1$
$40 \times 40$	$7.37 \times 10^{-3}$	1.99	$8.35 \times 10^{-1}$	$6.54 \times 10^{-2}$	1.78	$5.89 \times 10^0$
$80 \times 80$						

Table 6.3: 用移动网格计算碰撞流的误差,  $\nu = 1.0$ .

我们采用上面一章中的移动网格方法来求解这个算例。选取 (5.2) 为控制函数，其中  $\vec{u} = (u_x, u_y)^T$ .  $\alpha$  和  $\beta$  分别取为 0.002 和 2. 从 Table(6.3) 中可以看

网格	涡量 $L^2$ 误差	误差阶	散度 $L^2$ 误差	误差阶
$10 \times 10$	$2.35 \times 10^0$		$2.41 \times 10^0$	
$20 \times 20$	$1.16 \times 10^0$	1.01	$1.20 \times 10^0$	1.00
$40 \times 40$	$5.79 \times 10^{-1}$	1.00	$6.01 \times 10^{-1}$	1.00
$80 \times 80$				

Table 6.4: 用移动网格计算碰撞流的散度和涡量误差,  $\nu = 1.0$ .

出速度  $L^2$  误差有二阶, 压力收敛阶是一阶。从 Table(6.4) 可以看出速度散度和涡量均有一阶收敛。

首先我们判断网格是不是往正确的方向移动。从 Figure(6.1) 中看出控制函数  $G_1$  最大的地方分布在区域的四个顶角上, 中间区域控制函数的值是比较小的。网格应该从控制函数小的地方移动到控制函数值大的地方。而网格的移动方向也确实是在往四个顶角上移动。因此我们可以确定网格移动方向是对的。再者, 判断选取  $G_1$  为控制函数是否合适。从 Figure (6.2) 中看到: 在均匀网格下, 速度  $L^2$  误差最大的地方分布在四个顶角上, 这与控制函数的分布是一致的。所以选取  $G_1$  为控制函数是合理的。再来对比 Table(6.1) 和 Table(6.3) 中速度和压力的误差。可以发现, 移动网格方法下速度和压力的误差均有所下降, 但是下降的很小。

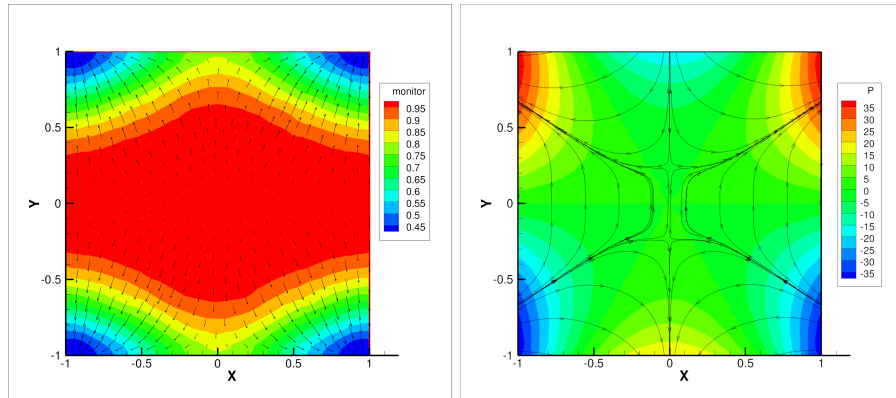


Figure 6.1: 左:  $m = \frac{1}{G_1}$  的等高线和网格移动方向; 右: 压力等高线和速度的流速线  $\alpha = 0.002, \beta = 2.0$ .

注意到在 Figure(6.3) 中, 移动网格相对于均匀网格移动的并不是很明显。在 Table(6.5) 中我们选取不同的  $\alpha$  的值, 查看速度和压力误差的变化。网格如 Figure(6.4) 所示。随着  $\alpha$  的值变大, 虽然网格移动的效果越来越明显, 但此时的速度和压力误差却变大了。

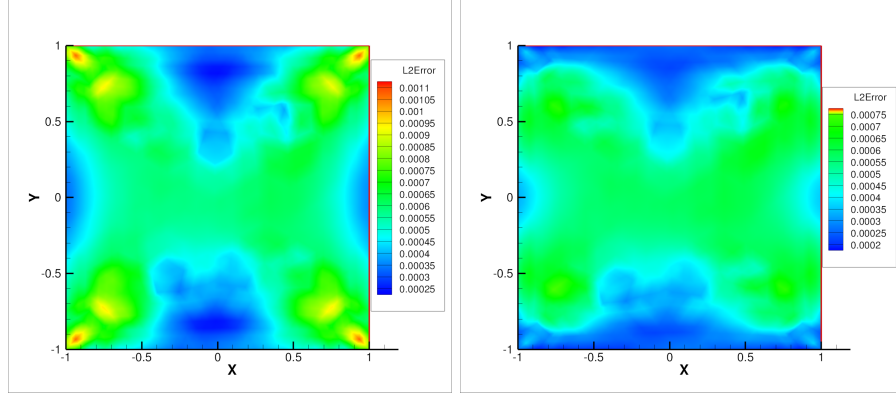


Figure 6.2: 速度  $L^2$  误差分布, 左: 均匀网格; 右: 移动网格  $\alpha = 0.002, \beta = 2.0$ .

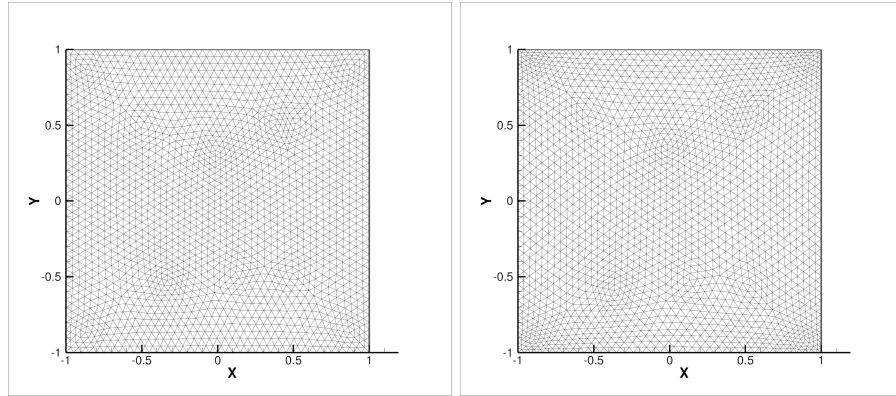


Figure 6.3: 网格对比, 左: 均匀网格  $20 \times 20$ ; 右: 移动网格  $\alpha = 0.002, \beta = 2.0$ .

$\alpha$	$\ \vec{u} - \vec{u}_h\ _{L^2}$	$\ \vec{u} - \vec{u}_h\ _{H^1}$	$\ p - p_h\ _{L^0}$	$\ p - p_h\ _{H^1}$
0.0(不移动)	$3.54 \times 10^{-2}$	$1.81 \times 10^0$	$3.90 \times 10^{-1}$	$1.22 \times 10^1$
0.002	$2.94 \times 10^{-2}$	$1.67 \times 10^0$	$2.330 \times 10^{-1}$	$1.06 \times 10^1$
0.005	$2.98 \times 10^{-2}$	$1.64 \times 10^0$	$2.08 \times 10^{-1}$	$1.06 \times 10^1$
0.01	$3.22 \times 10^{-2}$	$1.65 \times 10^0$	$2.02 \times 10^{-1}$	$1.08 \times 10^1$
0.1	$5.28 \times 10^{-2}$	$2.09 \times 10^0$	$3.39 \times 10^{-1}$	$1.40 \times 10^1$

Table 6.5: 不同  $\alpha$  的值, 用移动网格计算碰撞流的误差, 网格  $20 \times 20$ .

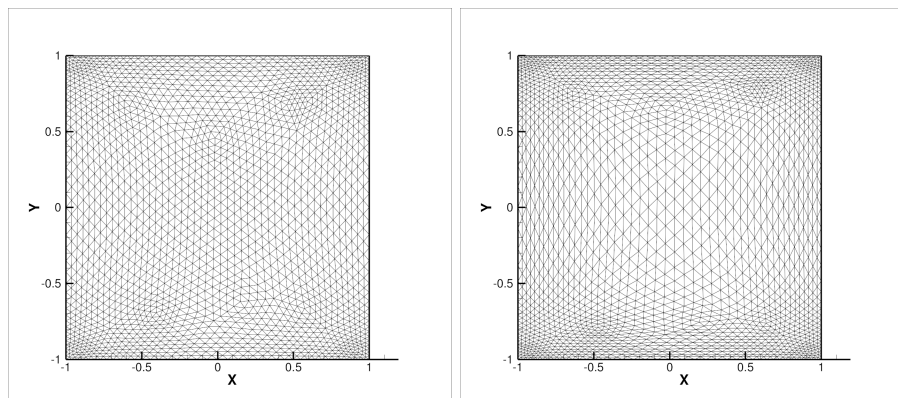


Figure 6.4: 网格对比, 左: 移动网格  $\alpha = 0.01, \beta = 2.0$  右: 移动网格  $\alpha = 0.1, \beta = 2.0$ .

## Bibliography

- [1] Ionut Danaila, Raluca Moglan, Frédéric Hecht, and Stéphane Le Masson. A newton method with adaptive finite elements for solving phase-change problems with natural convection. *Journal of Computational Physics*, 274:826–840, 2014.
- [2] Mohamed S Ebeida, Roger L Davis, and Roland W Freund. Unsteady incompressible flow simulation using galerkin finite elements with spatial/temporal adaptation. In *47th AIAA Aerospace Sciences Meeting, Orlando, FL*, 2009.
- [3] Stefano Berrone and Massimo Marro. Space–time adaptive simulations for unsteady navier–stokes problems. *Computers & Fluids*, 38(6):1132–1144, 2009.
- [4] Haibiao Zheng, Yanren Hou, and Feng Shi. A posteriori error estimates of stabilization of low-order mixed finite elements for incompressible flow. *SIAM Journal on Scientific Computing*, 32(3):1346–1360, 2010.
- [5] Yana Di, Ruo Li, Tao Tang, and Pingwen Zhang. Moving mesh finite element methods for the incompressible navier–stokes equations. *SIAM Journal on Scientific Computing*, 26(3):1036–1056, 2005.
- [6] Michel Bercovier and Olivier Pironneau. Error estimates for finite element method solution of the stokes problem in the primitive variables. *Numerische Mathematik*, 33(2):211–224, 1979.
- [7] Ruo Li. On multi-mesh h-adaptive methods. *Journal of Scientific Computing*, 24(3):321–341, 2005.
- [8] Max D Gunzburger. *Finite Element Methods for Viscous Incompressible Flows: A guide to theory, practice, and algorithms*. Elsevier, 2012.